Integrating ACT-R with Task Network Models

Christian Lebiere

Carnegie-Mellon University

Joint work with Eric Biefeld (CMU), Laurel Allender and Troy Kelley (ARL), Rick Archer and Dan Schunk(MA&D)

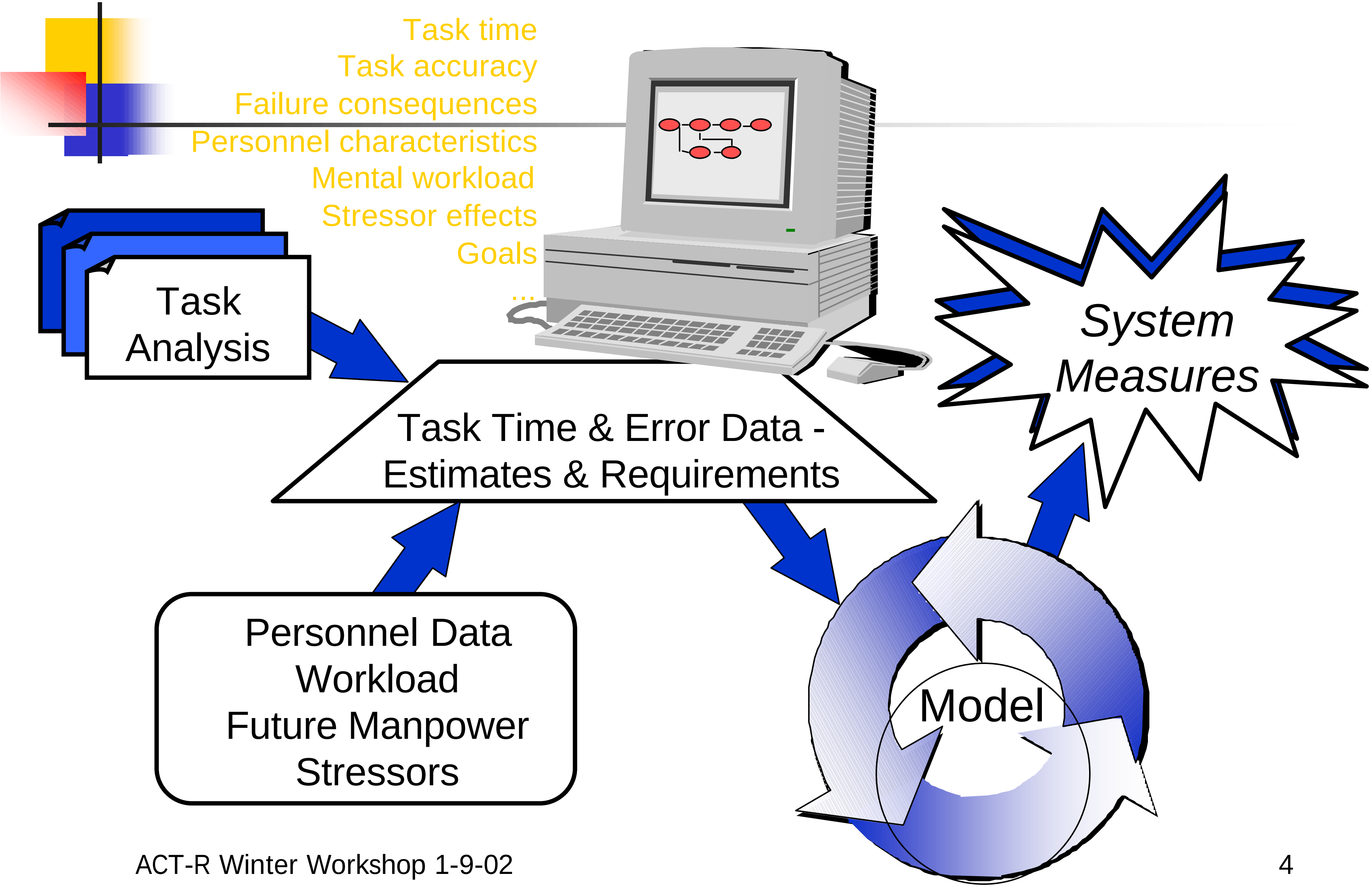
Motivation

- Make ACT-R accessible to non-programmer and/or non-psychologist modelers (cf. GOMS)
- Integrate ACT-R with existing models built by domain experts using their own tool (ie. implantable head into foreign bodies)
- Extend ACT-R with powerful simulation tool
- Bootstrap ACT-R into new domains
- Remedy ACT-R weakness above unit task level
- Achieve efficiency and scaling by providing cognitive accuracy only where needed

What is IMPRINT?

- Owned by ARL and developed by MA&D
- Based on discrete-event network simulation development tool (MicroSaint)
- Augmented with knowledge of human performance to model manpower and other Army needs

IMPRINT Architecture



Task time
Task accuracy
Failure consequences
Personnel characteristics
Mental workload
Stressor effects
Goals
...

Task Analysis

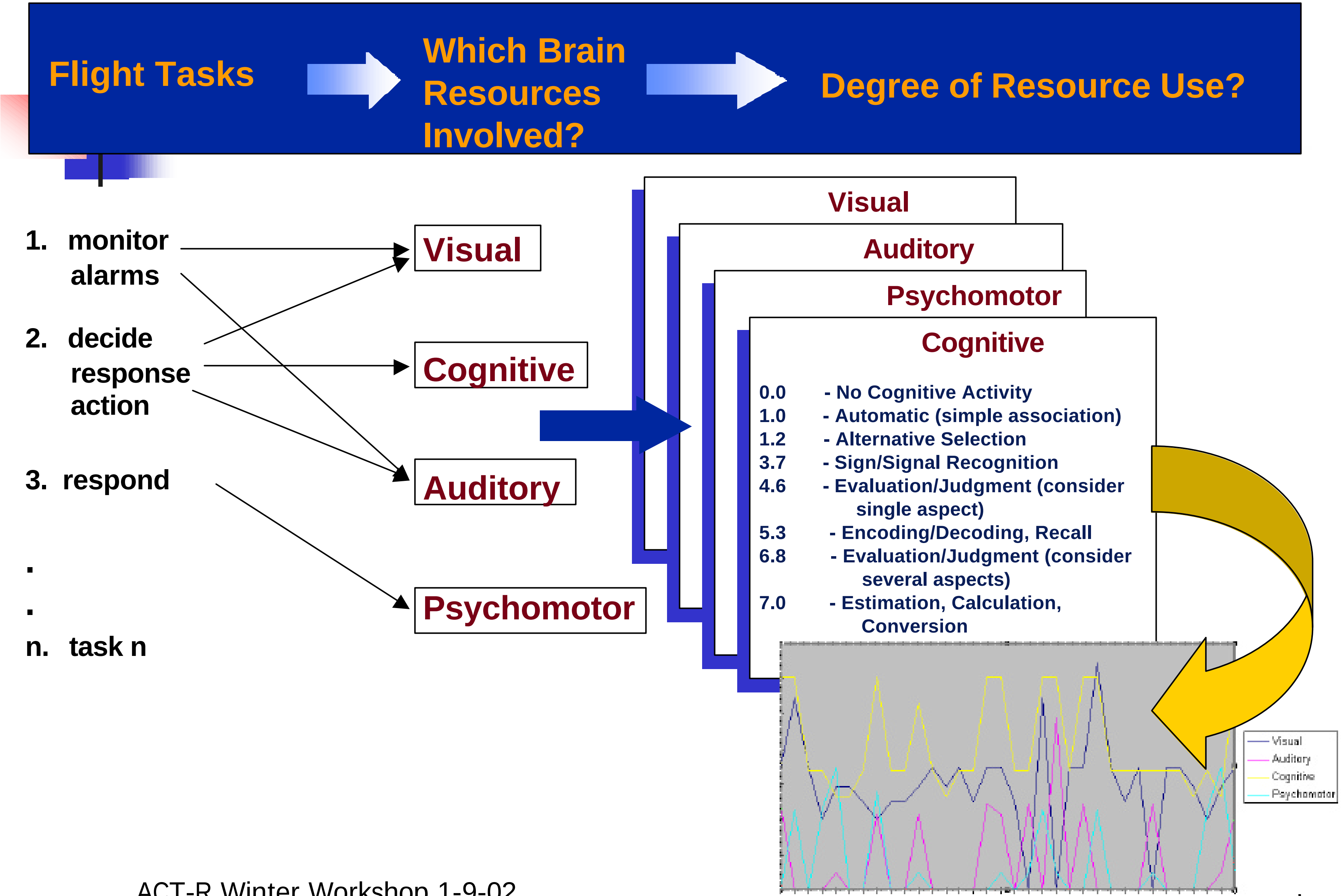
Task Time & Error Data - Estimates & Requirements

System Measures

Personnel Data
Workload
Future Manpower
Stressors

Model

Mental Workload



Flight Tasks → Which Brain Resources Involved? → Degree of Resource Use?

1. monitor alarms
2. decide response action
3. respond

.

.

n. task n

Visual

Cognitive

Auditory

Psychomotor

Visual

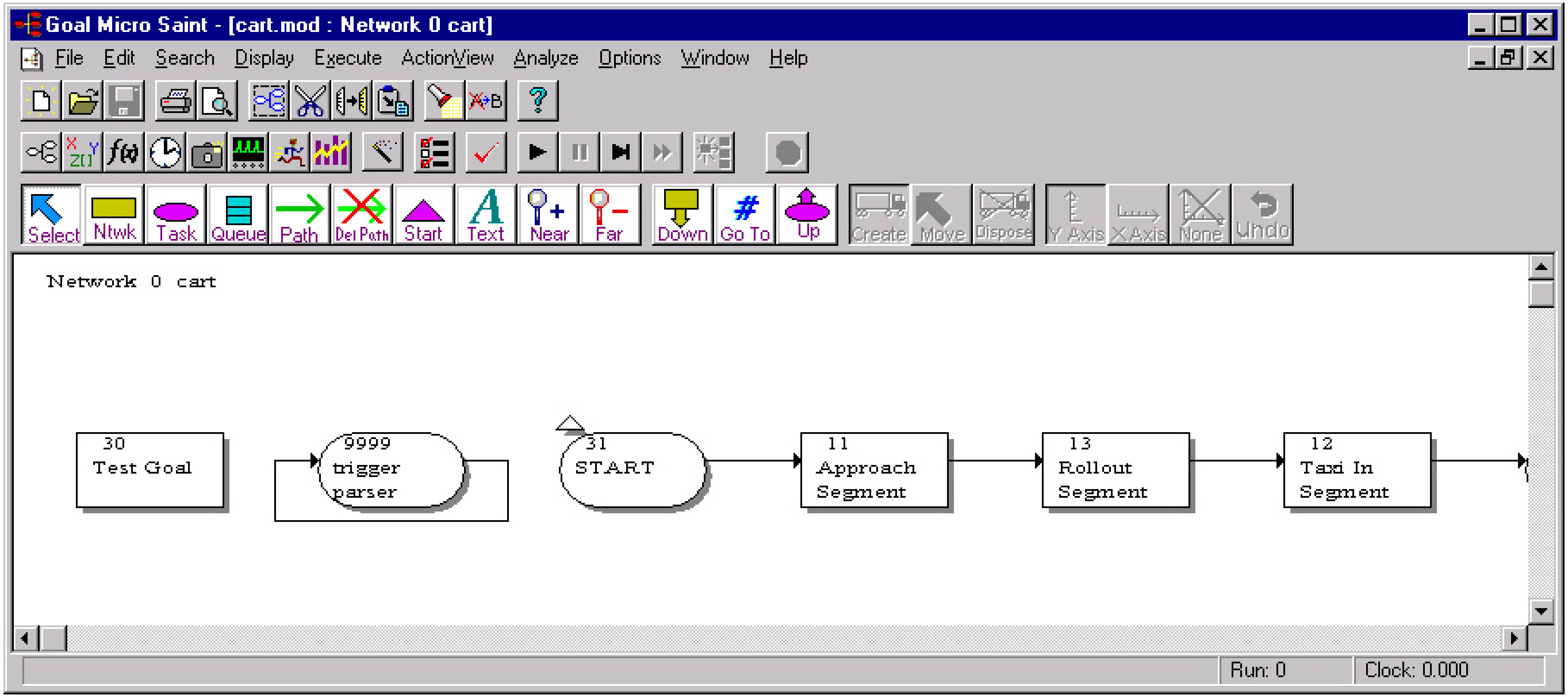
Auditory

Psychomotor

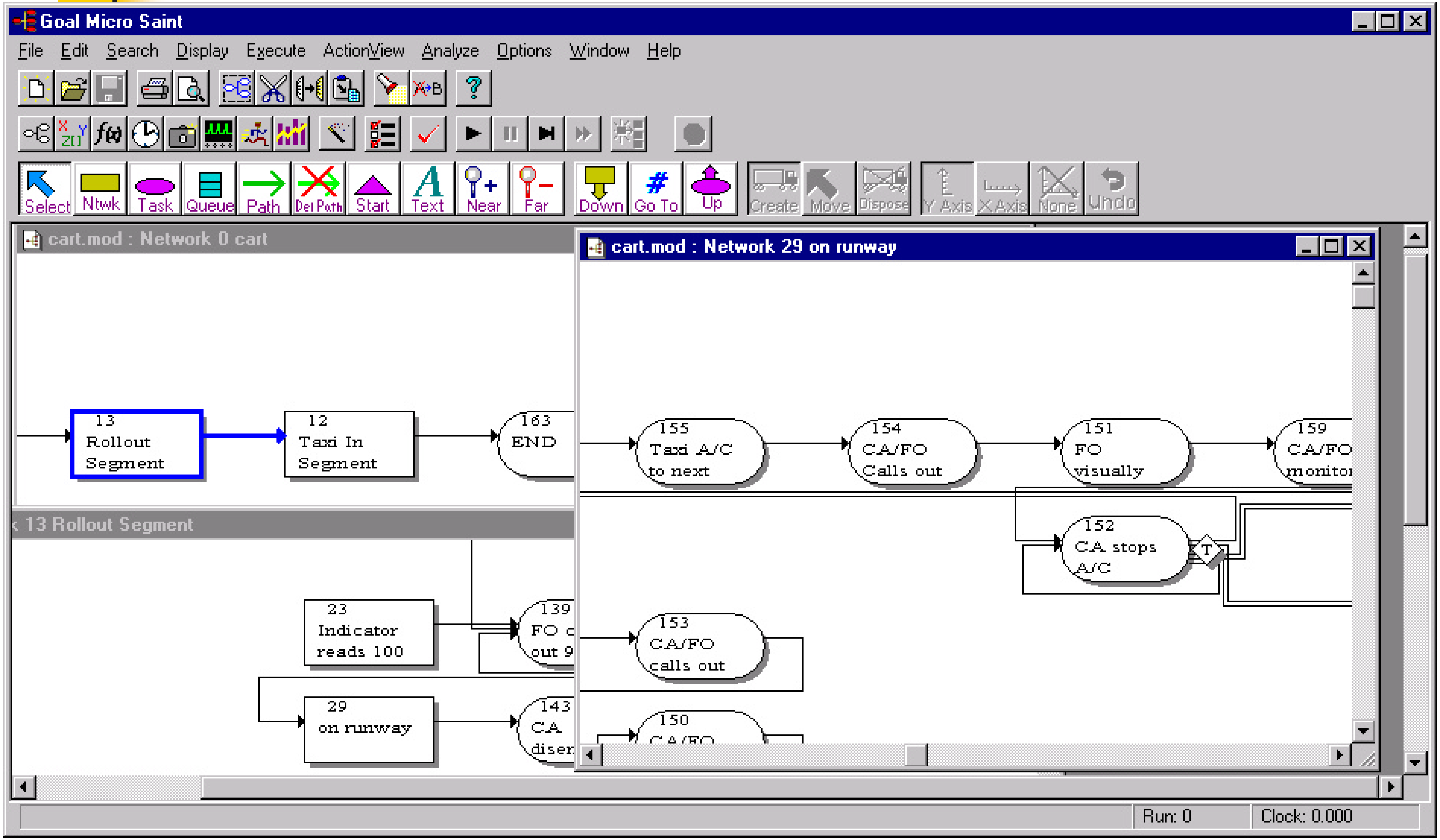
Cognitive

0.0	- No Cognitive Activity
1.0	- Automatic (simple association)
1.2	- Alternative Selection
3.7	- Sign/Signal Recognition
4.6	- Evaluation/Judgment (consider single aspect)
5.3	- Encoding/Decoding, Recall
6.8	- Evaluation/Judgment (consider several aspects)
7.0	- Estimation, Calculation, Conversion

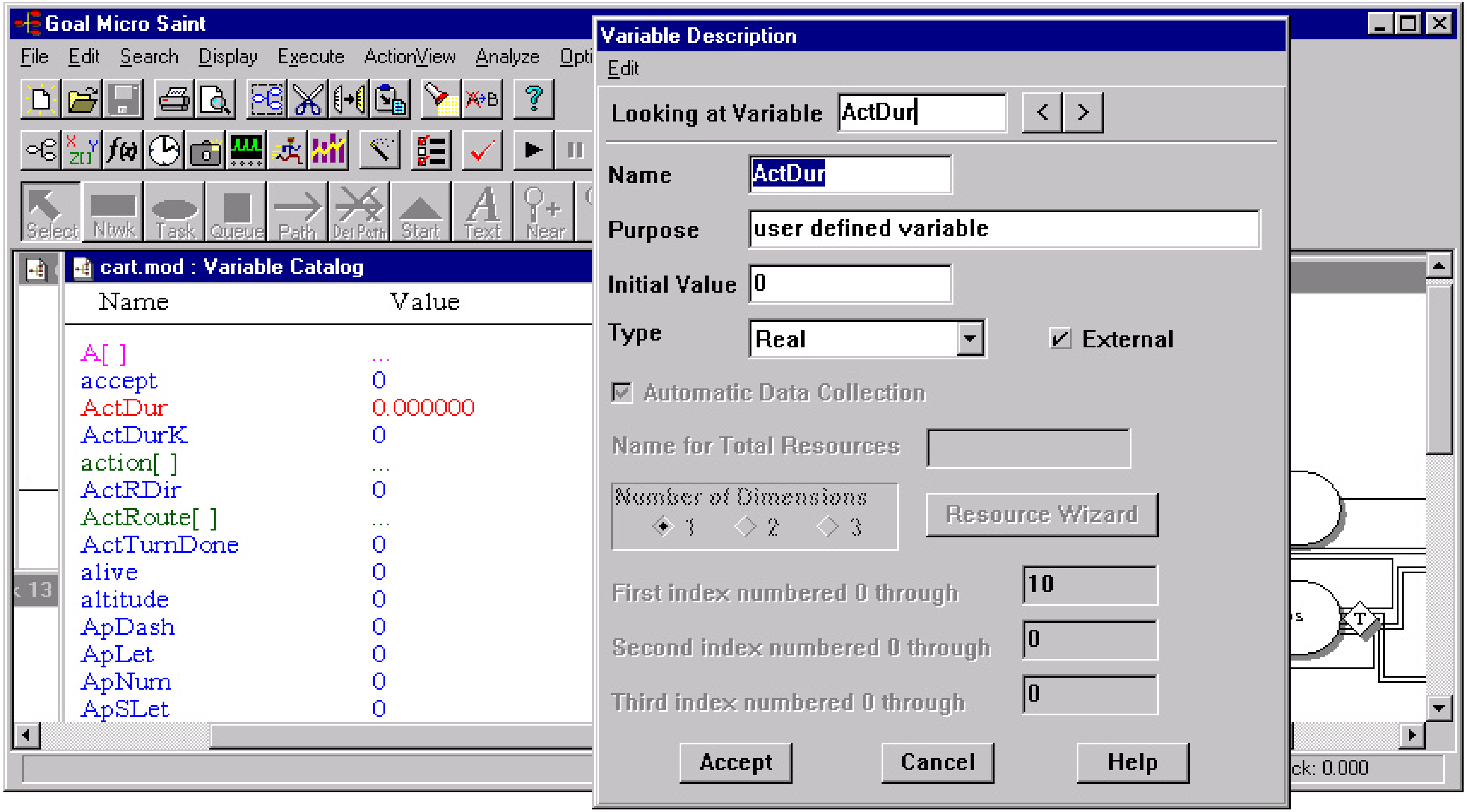
Goal Micro Saint GUI


Goal Micro Saint - [cart.mod : Network 0 cart]
File Edit Search Display Execute ActionView Analyze Options Window Help
Select Ntwk Task Queue Path Del Path Start Text Near Far Down Go To Up Create Move Dispose Y Axis X Axis None Undo
Network 0 cart
30 Test Goal
9999 trigger parser
31 START
11 Approach Segment
13 Rollout Segment
12 Taxi In Segment
Run: 0
Clock: 0.000

Task Network



One Description at a Time


Goal Micro Saint
File Edit Search Display Execute ActionView Analyze Opti
Select Ntwk Task Queue Path Del Path Start Text Near
cart.mod : Variable Catalog
Variable Description
Edit
Resource Wizard
Accept
Cancel
Help
ck: 0.000

Name	Value
A[ ]	...
accept	0
ActDur	0.000000
ActDurK	0
action[ ]	...
ActRDir	0
ActRoute[ ]	...
ActTurnDone	0
alive	0
altitude	0
ApDash	0
ApLet	0
ApNum	0
ApSLet	0

Looking at Variable: ActDur

Name: ActDur

Purpose: user defined variable

Initial Value: 0

Type: Real

☑ External

☑ Automatic Data Collection

Name for Total Resources:

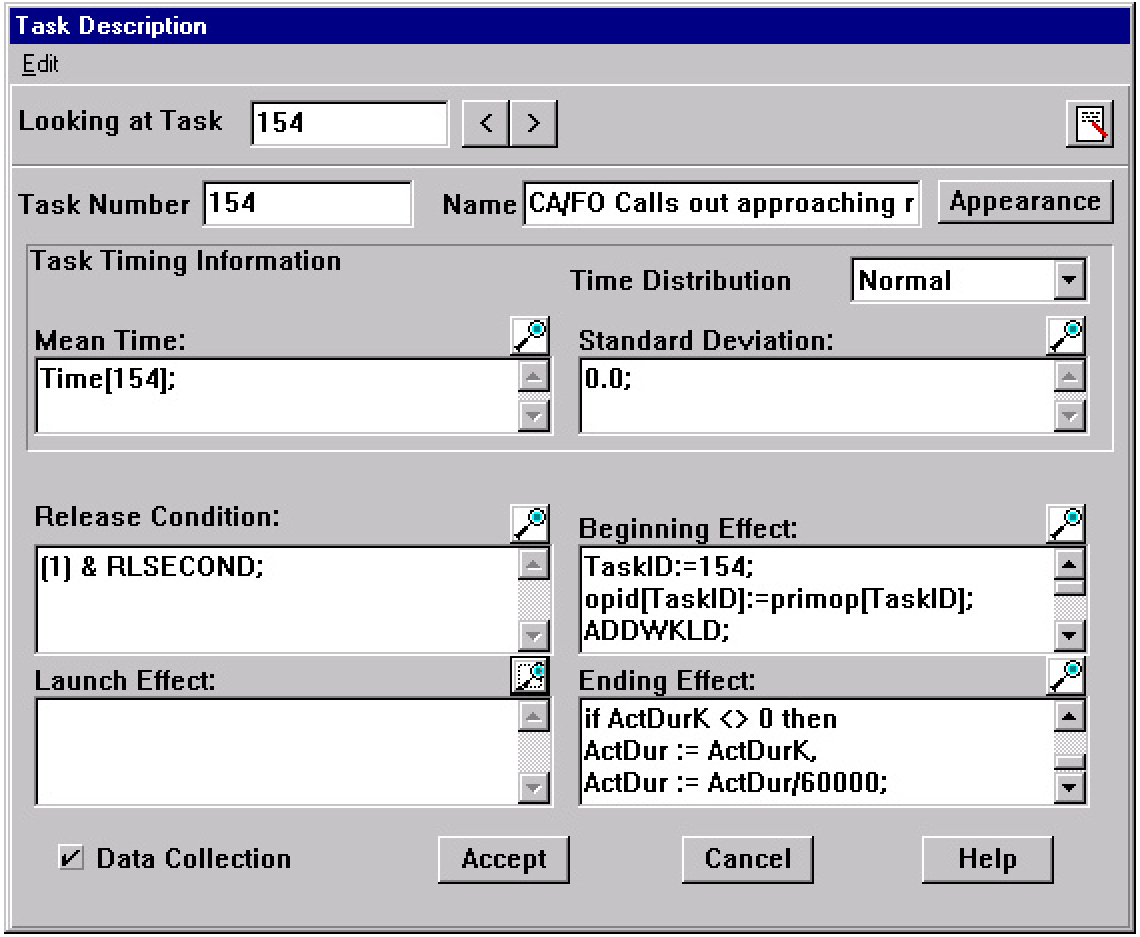
Number of Dimensions: ● 1 ○ 2 ○ 3

First index numbered 0 through: 10

Second index numbered 0 through: 0

Third index numbered 0 through: 0

Task Description



Task Description

Edit

Looking at Task 154

Task Number 154

Name CA/FO Calls out approaching r

Task Timing Information

Time Distribution Normal

Mean Time:

```
Time[154];
```

Standard Deviation:

```
0.0;
```

Release Condition:

```
[1] & RLSECOND;
```

Beginning Effect:

```
TaskID:=154;
opid[TaskID]:=primop[TaskID];
ADDWKLD;
```

Launch Effect:

Ending Effect:

```
if ActDurK <> 0 then
ActDur := ActDurK,
ActDur := ActDur/60000;
```

☑ Data Collection

- GUI is pretty
- When network is non trivial GUI gets very cluttered
- Pop up maze limits inspection
- Limits of built-in language
- Be glad ACT-R lives in LISP!

Integration Principles

- Modeling is modeling, whatever the idiom
- IMPRINT is built around the concept of tasks
- ACT-R is (?) built around the concept of goals
- Tasks roughly correspond to goals, which provide a natural level to perform the integration
- Expand some tasks that require cognitive accuracy into calls to ACT-R goals
- Variable descriptions of tasks are goal inputs
- ACT-R outputs back time to perform goal, decision/errors, perhaps workload estimate

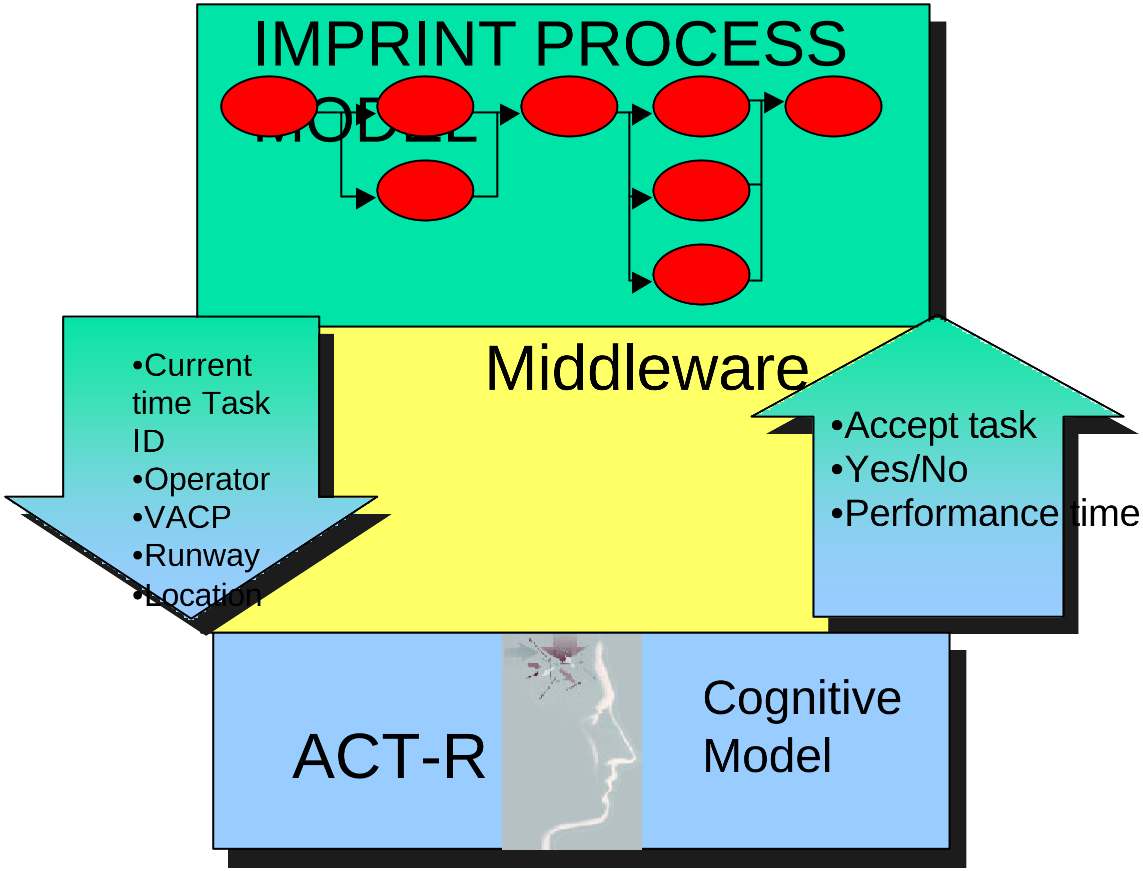
Application: Runway incursions

- (Used to be) FAA #1 problem
- Aircraft makes wrong turn during taxi
- Brain-dead procedure
  - No guidance from ATC during taxiing
  - Receives list of taxiways before landing
- Usually reported but can be corrected
- Most dangerous when poor visibility
  - Recent accident in Milan between GA and CA
- Funded by NASA ASP HEM project

Data Available

- About 100 runs on 9 routes
- A dozen errors with no discernible pattern
- Summary data from a bunch of papers
- Nominal Task Analysis
- Visual information (video, maps of O’Hare)
- Sample of communications
- Workload and Situation Awareness data
- Individual differences between pilots

IMPRINT/ACT-R Model



IMPRINT PROCESS MODEL

Middleware

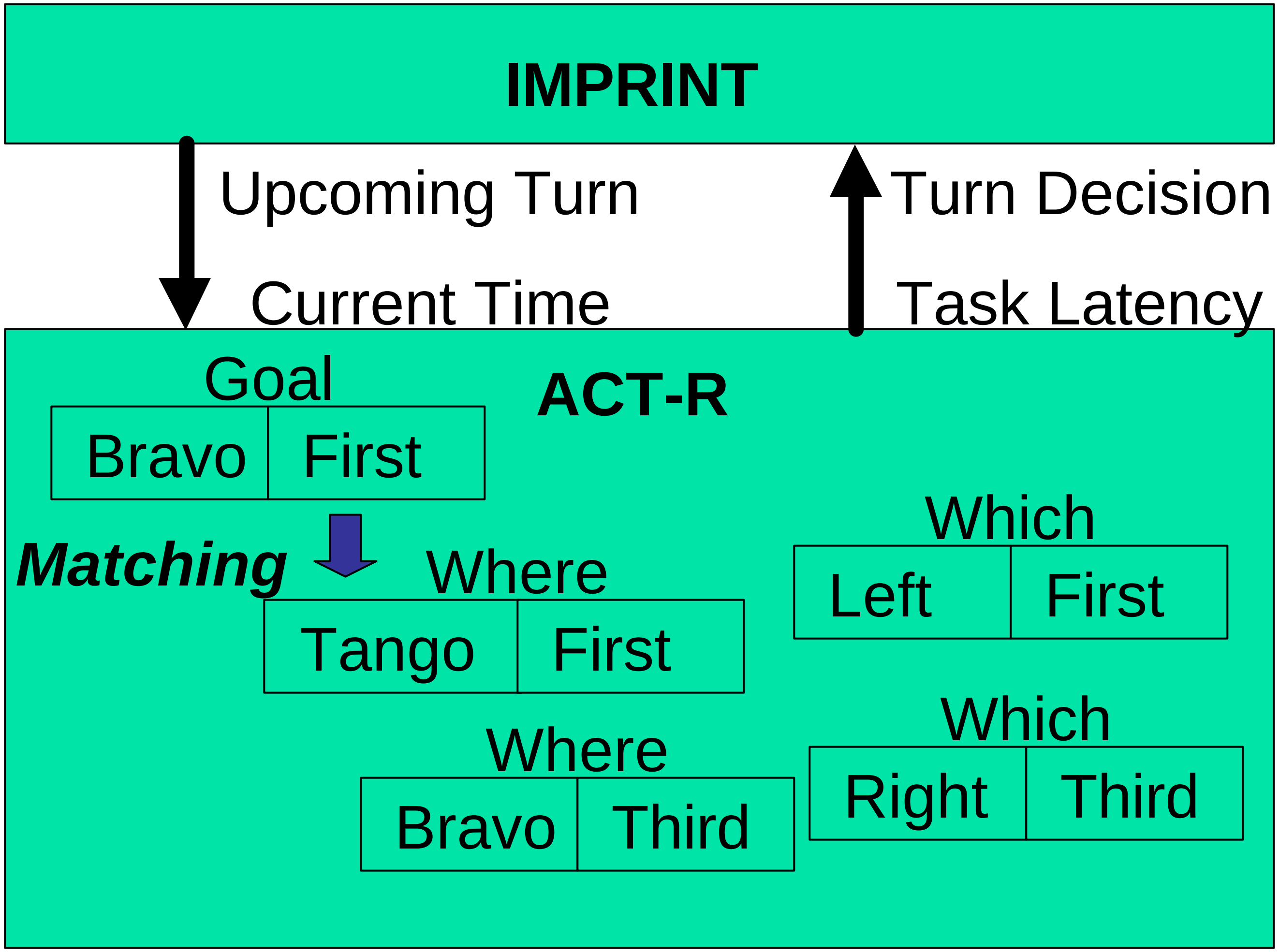
- Current time Task ID
- Operator
- VACP
- Runway
- Location

- Accept task
- Yes/No
- Performance time

ACT-R

Cognitive Model

ACT-R Model



IMPRINT

Upcoming Turn

Current Time

Turn Decision

Task Latency

ACT-R

Goal

Bravo	First

Matching

Where

Tango	First

Where

Bravo	Third

Which

Left	First

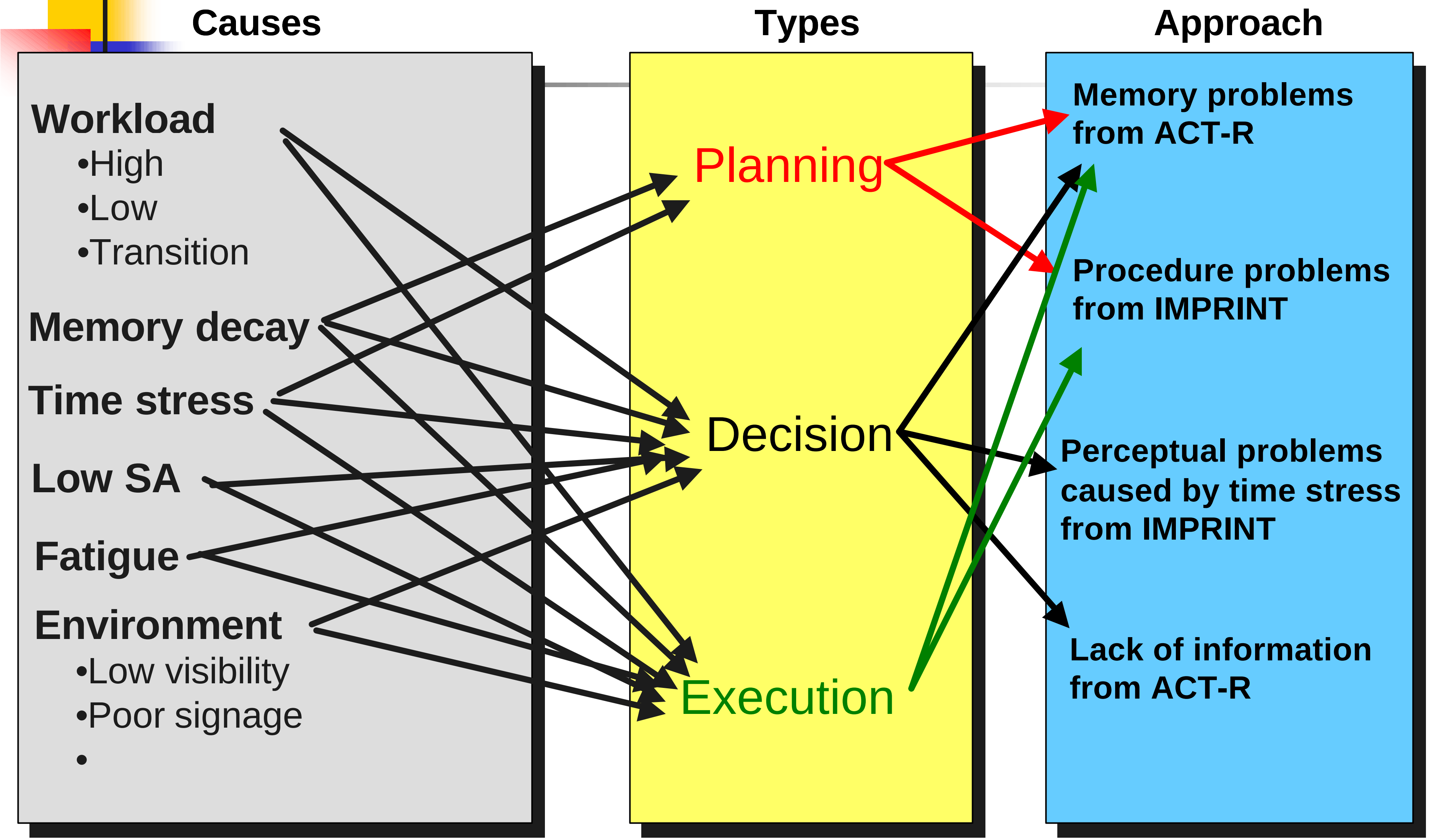
Which

Right	Third

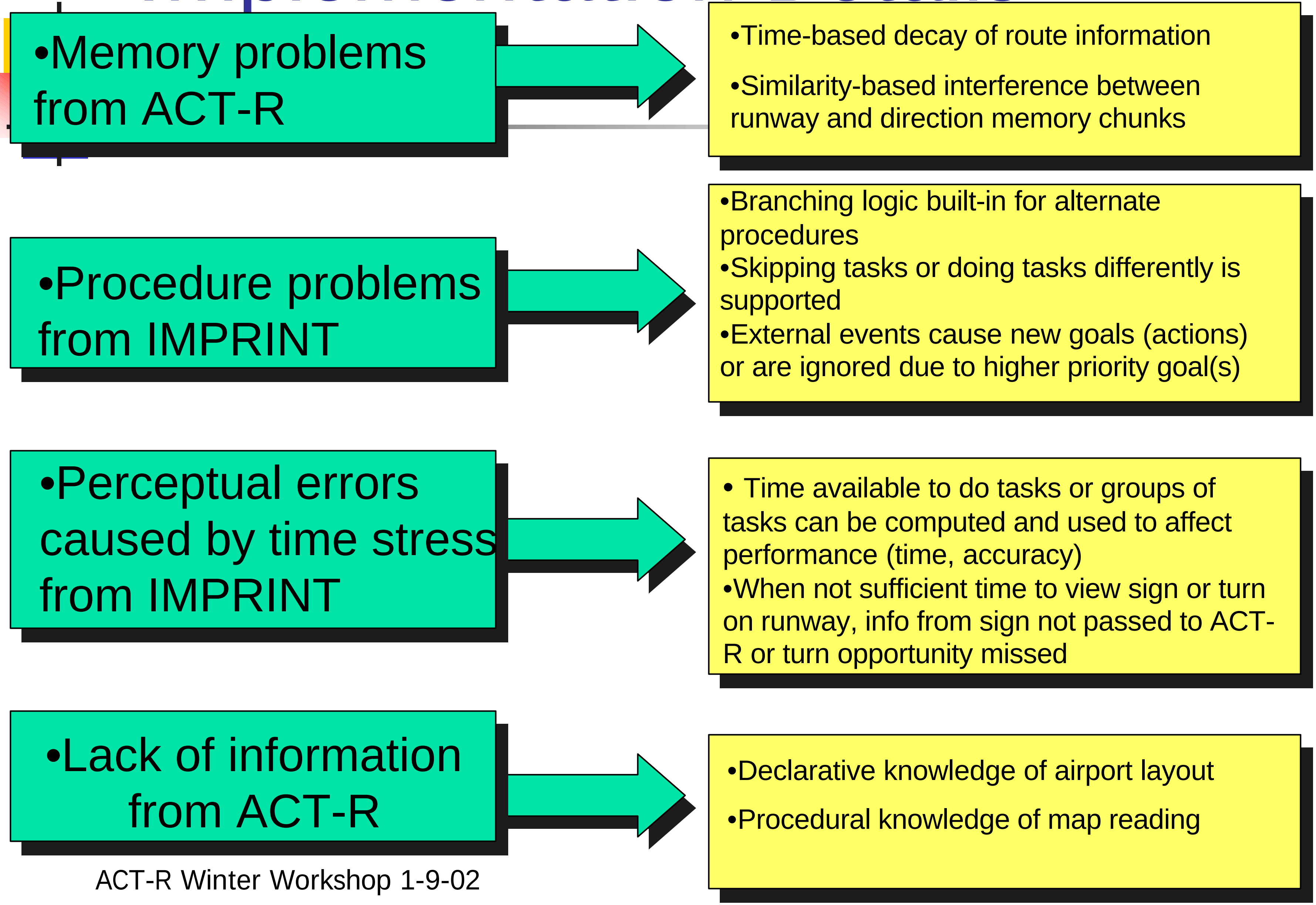
Memory Errors

- Omission
  - Forgot a where chunk (decay, noise)
  - Error: go straight instead of turn
- Commission 1
  - Remembers wrong runway (interference, similarity, priming, activation noise)
  - Error: turns on wrong runway or misses turn
- Commission 2
  - Remembers wrong turn (interference, noise)
  - Error: makes wrong turn on correct runway

Error Modeling Approach



Implementation Details



- Memory problems from ACT-R
  - Time-based decay of route information
  - Similarity-based interference between runway and direction memory chunks

- Procedure problems from IMPRINT
  - Branching logic built-in for alternate procedures
  - Skipping tasks or doing tasks differently is supported
  - External events cause new goals (actions) or are ignored due to higher priority goal(s)

- Perceptual errors caused by time stress from IMPRINT
  - Time available to do tasks or groups of tasks can be computed and used to affect performance (time, accuracy)
  - When not sufficient time to view sign or turn on runway, info from sign not passed to ACT-R or turn opportunity missed

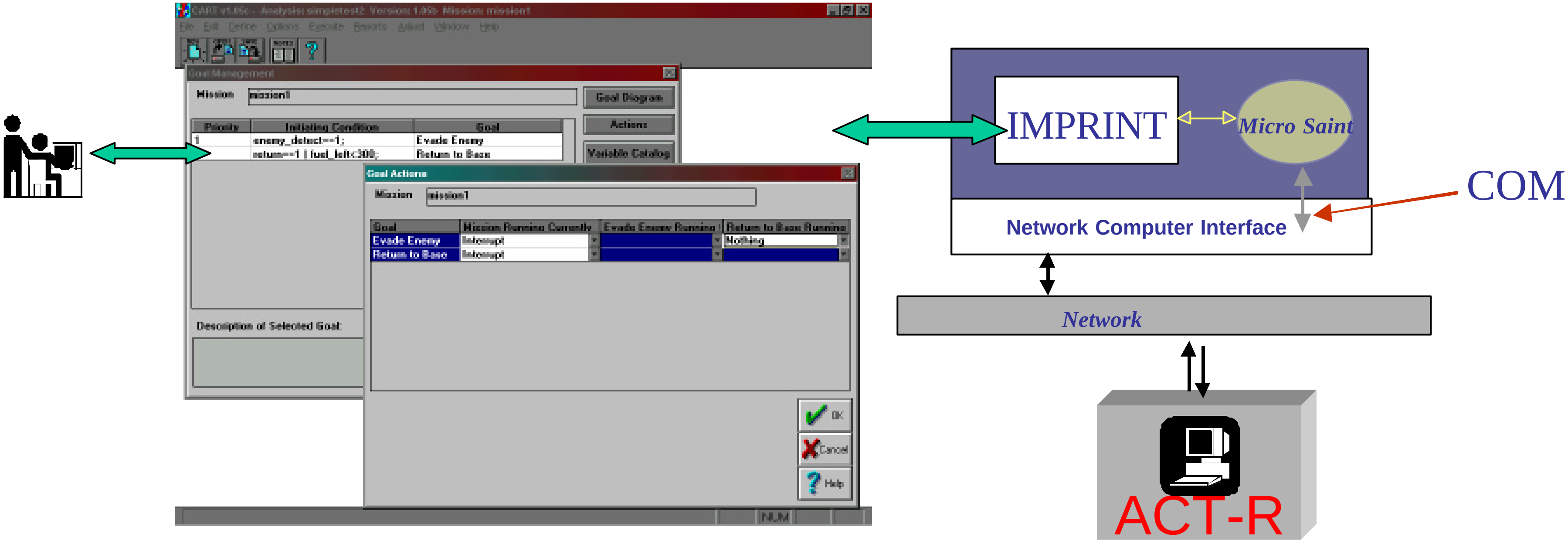
- Lack of information from ACT-R
  - Declarative knowledge of airport layout
  - Procedural knowledge of map reading

Scenario Events

- Using O’Hare maps
  - Calculated estimated time between runway turnoffs
    - distance between runway turns x assumed ground speed
  - Calculated estimated time available to view sign
    - distance to signage x assumed ground speed
- Times checked with video
  - Our calculations resulted in shorter times, possibly due to less conservative ground speed estimates
- Events used by IMPRINT to cause tasks to be triggered

Interoperability Features

- Component Object Model enabled
- External variables
- External application calls



CART v1.05c - Analysis: simpletest2 Version: 1.05b Mission: mission1

File Edit Define Options Execute Reports Adjust Window Help

Goal Management

Mission: mission1

Priority	Initiating Condition	Goal
1	enemy_detect==1;	Evade Enemy
	return==1 || fuel_left<300;	Return to Base

Goal Diagram | Actions | Variable Catalog

Description of Selected Goal:

Goal Actions

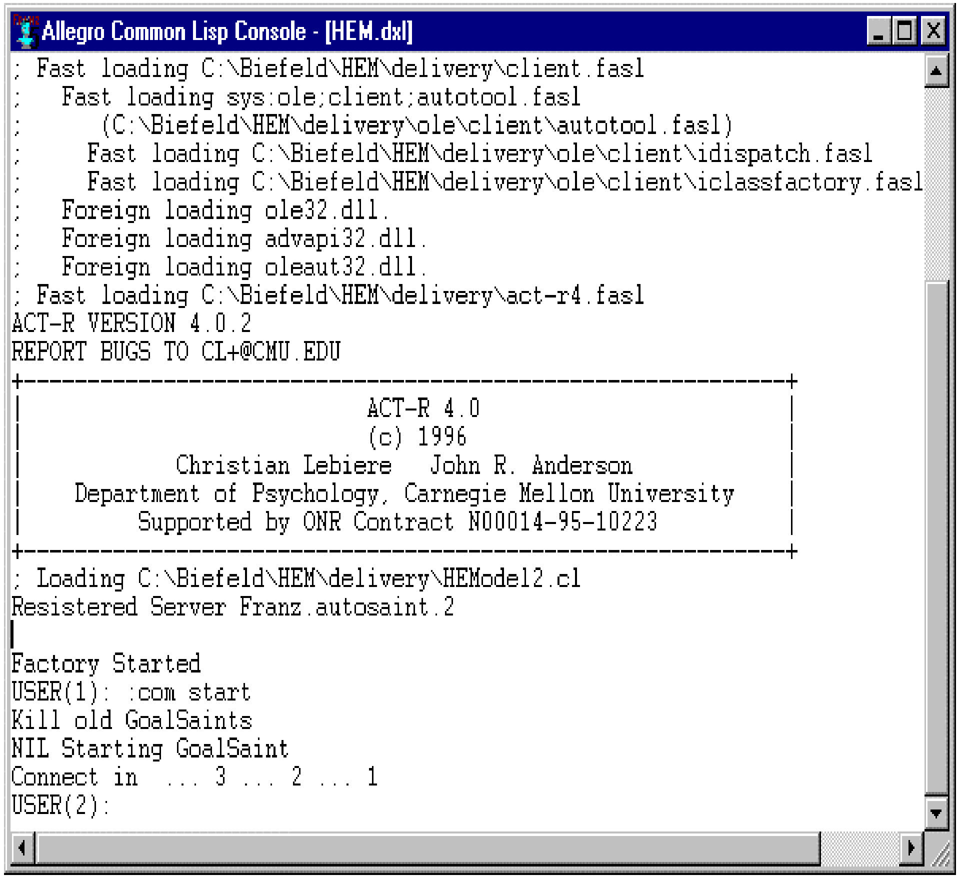
Mission: mission1

Goal	Mission Running Currently	Evade Enemy Running	Return to Base Running
Evade Enemy	Interrupt		Nothing
Return to Base	Interrupt		

OK | Cancel | Help

NUM

Started (and Ended) by LISP



```
; Fast loading C:\Biefeld\HEM\delivery\client.fasl
;   Fast loading sys:ole;client;autotool.fasl
;      (C:\Biefeld\HEM\delivery\ole\client\autotool.fasl)
;     Fast loading C:\Biefeld\HEM\delivery\ole\client\idispatch.fasl
;     Fast loading C:\Biefeld\HEM\delivery\ole\client\iclassfactory.fasl
;   Foreign loading ole32.dll.
;   Foreign loading advapi32.dll.
;   Foreign loading oleaut32.dll.
; Fast loading C:\Biefeld\HEM\delivery\act-r4.fasl
ACT-R VERSION 4.0.2
REPORT BUGS TO CL+@CMU.EDU
+--------------------------------------------------------------+
|                            ACT-R 4.0                         |
|                            (c) 1996                          |
|               Christian Lebiere    John R. Anderson           |
|          Department of Psychology, Carnegie Mellon University |
|               Supported by ONR Contract N00014-95-10223       |
+--------------------------------------------------------------+
; Loading C:\Biefeld\HEM\delivery\HEModel2.cl
Resistered Server Franz.autosaint.2

Factory Started
USER(1): :com start
Kill old GoalSaints
NIL Starting GoalSaint
Connect in  ... 3 ... 2 ... 1
USER(2):
```

- Just launch LISP application
- Use top level commands :COM
- LISP loads OLE tools and starts GoalSaint.Exe

Typical results

```
 Run: 5

OK   #: 1 Turn: M6 Route: c2r Direction: straight Route straight
OK   #: 2 Turn: n2 Route: c2r Direction: straight Route straight
OK   #: 3 Turn: c2r Route: c2r Direction: left Route left
OK   #: 4 Turn: c3 Route: h7 Direction: straight Route straight
OK   #: 5 Turn: h7 Route: h7 Direction: right Route right
OK   #: 6 Turn: b1 Route: b4 Direction: straight Route straight
OK   #: 7 Turn: h6r Route: b4 Direction: straight Route straight
OPPS #: 8 Turn: b4 Route: b4 Direction: right Route left
[1c] USER(21): :res
```

- Sample of 5 runs: 2 correct and 3 errors
- 2 forgetting of turns, one turn the wrong way on the right runway

Lessons Learned

- Interface issues are very time-consuming
  - Though the second time seems much easier
- Difficult to develop both models in parallel
  - Need stubs, testing, documentation, access...
- Difference in paradigms is the biggest barrier
  - "Impedance mismatch problem"
- HLA wisdom: "You need an agent endowed with the wisdom of Solomon, the patience of Job, and the authority of Caesar."

Modeling Requirements

- Access to a simulation to run the model
  - We have it but you don’t need it...
  - Task network model as simulation?
- Simulation must run fast and in batch
  - Almost all CGF simulations are hand-operated
  - Real-time good enough for HITL so...
- Access to detailed data is essential
  - We have it but you can’t have it...
  - We had it but we debriefed then erased it…

Current & Future Applications

- Integrating ACT-R into Combat Automation Requirements Testbed (CART) model (AF)
- Second phase of HEM project: full-flight with Synthetic Vision System (SVS)
- Model Army commanders decision-making (e.g. BC2010 desktop training simulation)
- Some outstanding integration issues:
  - Parallelism: reconcile ACT-R and IMPRINT views
  - Workload: expand and refine ACT-R definition
  - Possible integration of tools and interfaces?

Motivation Revisited

- Make ACT-R accessible to non-programmer and/or non-psychologist modelers (cf. GOMS)
- Integrate ACT-R with existing models built by domain experts using their own tool (ie. implantable head into foreign bodies)
- Extend ACT-R with powerful simulation tool
- Bootstrap ACT-R into new domains
- Remedy ACT-R weakness above unit task level
- Achieve efficiency and scaling by providing cognitive accuracy only where needed

If not this, then what?

- Reimplement every simulation in Lisp ...
- C or Java will melt all problems away ...
- Graphical interface to production authoring, other GUI tools (but PGSS 2001 warnings) ...
- General-purpose interface to HLA since it is becoming the interface standard ...
- Provide packaged functionality, i.e. function calls to memory(...), pattern-matching, etc